# Weighted Model Counting with Conditional Probabilities

Paulius Dilkas

19th July 2020

## 1 Introduction

**Legend.**

**F** Feedback (round 1).

**F2** Feedback (round 2).

**FP** Feedback from the review panel.

• My own idea.

**The big TODO list.**

- Look into ways to theoretically justify the performance benefits. Some suggestions:
  - the number of variables,
  - the number of clauses/CPTs/lines in CPTs,
  - the proportion of edges that exist,
  - average degree,
  - average number of clauses that a variable is in,
  - the number of clusters in the linear/tree-like decomposition.
- Have an example of how the ADDs function in this situation. If not for the paper, then at least for slides. Use the framework to check its correctness.

F Give a concrete example of something impossible to represent using WMC.

- Turn the saved PDFs into citations.

F What are the main claims, what are the main takeaways, intuitive [???] of theorems to follow. To do this, we appeal to algebraic constructions to define the main concepts for introducing measures on Boolean algebras.

F Can you say something here about factorized vs non-factorized weight function definitions? That is, factorized is when $w$ maps literals to $R_{\geq 0}$, non-factorized is when $w$ maps models to $R_{\geq 0}$ and

  - come up with nice example when non-factorized weights are intuitive;
  - clarify that the factorized definition have is w.r.t. models, in case some one gets confused. [It doesn't have to be, if the BA is not free—P.]

F2 The paper at this stage is very technical—the danger is that WMC/SRL people may not be able to follow it and so would be hard to get accepted. Without clear target audience, [???] get work accepted. My main high level suggestion is that let us tease apart what you have and see if a story emerges. That is, let us attempt to write a paper **with examples** and see whether with significant motivation, we have a story emerging. Below, sample text needed to adequately motivate [???] for WMC/SRL community.

FP Clear up questions about independence between literals.

F2 Preliminaries: explain with examples: models = elements [atoms] of algebra.

- WMC as a Measure

  F2 Need to explain how WMC and NWMC connects to standard definitions of WMC and $\Pr(\phi \mid e) = \mathrm{WMC}(\phi \wedge e) / \mathrm{WMC}(e)$.

  F2 You need to explain what precisely these mean in logic and models and weight functions are usually defined and understood.

  – Be careful about mentioning ideals, filters, and quotients.

- What Measures are WMC-Definable: proofs need to be updated and propositions could be phrased in a better way, but the gist should be the same.

- Extending the Algebra.

  F2 If you prove (b) above, you could motivate why weights on literals is attractive and whether there is a way to augment the expressivity of WMC while still maintaining literal level weights. Hence this action.

  F2 You need to explain the significance of this result.

- Make up my mind about $a, b$ vs. $x, y$ and stick to it (maybe $x, y$?).

- Terminology: 'with generating set $S$' → 'over $S$'.

- Perhaps reorder the section of preliminaries into paragraphs, i.e., a paragraph for order, for homomorphisms, etc. This would take up less space.

- Notation: if $L$ denotes literals, then it doesn't denote a generating set.

- Prove that we have an algebra of functions with two (?) additional operations. Or maybe just cite the original paper for this.

- The function $\phi$ created by the algorithm can be seen as a measure on the BA $2^U$. Can I at least show that this BA has fewer elements as a measure of compactness w.r.t. the measure?

- Note that we're going to use logical notation for BAs. Give an example of the two notations.

- Could clarify why $[\lambda_{X=0}] = \overline{[\lambda_{X=1}]}$. This follows from the definition. Or should I avoid the overline notation?

- After the experiments are finished, note the processor, memory per thread, and add the following acknowledgment. This work has made use of the resources provided by the Edinburgh Compute and Data Facility (ECDF) (`http://www.ecdf.ed.ac.uk/`).

**Experimental setup.**

- Algorithms[1]
  - ADDMC [9] (rediscovered the multiplicativity of BAs in different words) (with optimal settings)
  - Cachet [24]
  - c2d [8]
  - d4 [20] (closed source, boo!)
  - miniC2D [23]
- Encodings:
  - `-d02` [7]
  - `-sbk05` [25]
  - `-cd05` [3]
  - `-cd06` [4] (supposed to be the best)
  - mine

  All except mine are from Ace 3.0[2] and should be compiled with `-encodeOnly` (i.e., don't compile the CNF into an AC) and `-noEclause` (i.e., only use standard syntax) flags.
- Datasets[3]
  - binary Bayesian networks from Sang et al.[4] [25]
    * Grid (networks) (ratio 75 means that 75% of the nodes are deterministic),
    * Plan recognition (problems),
    * Deterministic quick medical reference (what do the numbers mean? the README doesn't say).
  - Bayesian networks available with Ace
    * `2004-pgm` [6] (binary)
    * `2005-ijcai` [3]. The Genie/Smile files have their own citation data that I should probably extract. This is the only dataset that has some non-binary networks.
    * `2006-ijar` [6] (binary)
  - ProbLog [10]
  - probabilistic programs [16]

**Contributions.**

- WMC defines a measure over a BA.
- WMC with weights on literals imposes an independence assumption. (Measures are 'slightly' more expressive than WMC with weights on models because they apply to non-atomic BAs.)
- A BA can be augmented with new literals in order to support any measure.
- (Maybe) a lower bound on the number of new literals needed in order to support any measure.

[1] `http://beyondnp.org/pages/solvers/model-counters-exact/`
[2] `http://reasoning.cs.ucla.edu/ace/`
[3] There might be more at `https://meelgroup.github.io/`.
[4] `https://www.cs.rochester.edu/u/kautz/Cachet/`

- Alternatively, one can use coproducts and pushouts to define a BA with precisely the right independence and conditional independence conditions. (This requires a relaxed version of WMC.)
- This results in a smaller problem for WMC algorithms (w.r.t. both the number of literals and the length of the theory) and is optimal for, e.g., Bayesian networks.
- (Maybe) this results in faster inference (?)

**Notable previous/related work.**

- Hailperin's approach to probability logic [15]
- Nilsson's (somewhat successful) probabilistic logic [22]
- Logical induction: a big paper with a good overview of previous attempts to assign probabilities to logical sentences in a sensible way [13]
- Measures on Boolean algebras
  - On possibility and probability measures in finite Boolean algebras [2]
  - Representation of conditional probability measures [19]

**Notes.**

- Thesis: many important computational problems are solved by encoding them as WMC. But if the weight function is extended to 'conditional probabilities', the problem becomes easier because current approaches need workarounds in order to be independent.
- Shorter thesis: many important problems are encoded as WMC, but they can be encoded in a better way if we allow for conditional probabilities.
- We extend the Gaifman graph to add edges when two variables occur in the same CPT (e.g., including the edge from $A$ to $B$ when the CPT is $\Pr(A \mid B)$).
- Extra benefit: one does not need to come up with a way to turn some probability distribution to into a fully independent one.
- Trivial CPTs such as $\Pr(A \mid B) = \Pr(\neg A \mid B) = 0.5$, the ADDs of which simplify to a single number, are put in the first cluster.
- Observation: by inspecting the BN, we could identify CPTs that could be completely ignored, but maybe good heuristics take care of that anyway.
- Observation: ADDs have a restrict command. I'm not going to use it, but it could be nice.
- Alternative scenario to consider for future work: #SAT solver generates solutions which are then used to restrict the ADDs that calculate probabilities.
- Important future work: replacing ADDs with AADDs[5] is likely to bring performance benefits. Other extensions:
  - FOADDs can represent first order statements;
  - XADDs can replace WMI for continuous variables;
  - ADDs with intervals can do approximations.

[5] `https://github.com/ssanner/dd-inference`

- cd06 is supposed to be the fastest, but seems to be one of the slowest (along with cd05). Perhaps the best encoding for arithmetic circuits has very little to do with the best encoding for CNF WMC solvers?
- When the Bayesian network has an evidence file, we compute the probability of evidence. Otherwise, let $X$ denote the last-mentioned node in the Bayesian network. If `true` is a valid value of $X$, we compute the marginal probability of $X = \texttt{true}$. Otherwise, we pick the first value of $X$ and calculate its marginal probability. This applies to the Grid data set (as intended) and also to two instances of Plan Reconstruction and roughly half of the instances from 2004-PGM that have empty evidence files.
- Bayesian networks are often solved in a compile once, query many times fashion. This can be achieved using ADDMC by selecting a subset $S$ of variable we may want to query over and running ADDMC while excluding $S$ from variable elimination/projection/$\exists$.
- `cd05` relaxes the encoding so much that extra models become possible. They are supposed to be filtered out by the algorithm, but mine can't do that because it doesn't deal with models. Same for `cd06` because it's based on `cd05`.
- For ground ProbLog, we can encode a program

  ```
  p :: a :- b
  q :: a :- c
  ```

  into $P(a \mid b) = p$, $P(a \mid c) = q$ instead of having clauses $b \Rightarrow a$, $c \Rightarrow a$. Some logical structure is likely to remain.
- Potential criticism may be that this is a step backwards and doesn't allow us to use SAT-based techniques for probabilistic inference. However, they can still be used for the 'theory+query' part.
- We don't compare 'compile times' because our encoding time is linear, so we would easily beat everyone else.
- The claim behind this paper is that allowing for conditional probabilities in the context of weighted model counting seems to be a good idea. Bayesian networks and ADDMC are only particular examples. This should also work with Cachet.
- For the ProbLog $\rightarrow$ WMC conversion, check out this guy.[6]
- Zero-probability weights and one-probability weights can be interpreted as logical clauses. This doesn't affect ADDMC but could be useful for other solvers.
- Filtering out ADDs that have nothing to do with the answer helps tremendously, but I'm not sure if I should do that. Could a heuristic do the same thing?
- For each row in each CPT, we order values by decreasing probabilities. Need to check if this makes the solver more accurate.
- In the encoding:
  - We assume that the first literal after 'w' is positive.
  - The last two numbers are the positive and the negative probabilities, respectively—sometimes they add to one, and sometimes the negative probability is one, regardless of the value of the first probability.

[6] `https://users.ics.aalto.fi/ttj/`

- My encoding for non-binary BNs should still be faster because it doesn't have those 'equivalence' clauses.
- Guess: what makes the problem easier is:
  1. short clauses,
  2. low number of clauses.

  This explains why my first version of encoding non-binary CPTs was too slow.
- With all encodings (including mine non-binary encoding), they live on a measure Boolean algebra where the measure is not probabilistic (so $\Pr(\neg A) \neq 1 - \Pr(A)$).
- We assume that all variables in the Bayesian network have at least two values.
- Remark: for any function $A\colon 2^X \to \mathbb{R}_{\geq 0}$, $A + \overline{A} = 1$.

## 2 Preliminaries

**Definition 1.** A *Boolean algebra* (BA) is a tuple $(\mathbf{B}, \wedge, \vee, \neg, 0, 1)$ consisting of a set $\mathbf{B}$ with binary operations *meet* $\wedge$ and *join* $\vee$, unary operation $\neg$ and elements $0, 1 \in \mathbf{B}$ such that the following axioms hold for all $a, b, \in \mathbf{B}$:

- both $\wedge$ and $\vee$ are associative and commutative;
- $a \vee (a \wedge b) = a$, and $a \wedge (a \vee b) = a$;
- $0$ is the identity of $\vee$, and $1$ is the identity of $\wedge$;
- $\vee$ distributes over $\wedge$ and vice versa;
- $a \vee \neg a = 1$, and $a \wedge \neg a = 0$.

For clarity and succinctness, we will occasionally use three other operations that can be defined using the original three[7]:

$$
\begin{aligned}
a \to b &= \neg a \vee b, \\
a \leftrightarrow b &= (a \wedge b) \vee (\neg a \wedge \neg b), \\
a + b &= (a \wedge \neg b) \vee (\neg a \wedge b).
\end{aligned}
$$

We can also define a partial order $\leq$ on $\mathbf{B}$ as $a \leq b$ if $a = b \wedge a$ (or, equivalently, $a \vee b = b$) for all $a, b \in \mathbf{B}$. Furthermore, let $a < b$ denote $a \leq b$ and $a \neq b$. For the rest of this paper, let $\mathbf{B}$ refer to the BA $(\mathbf{B}, \wedge, \vee, \neg, 0, 1)$. For any $S \subseteq \mathbf{B}$, we write $\bigvee S$ for $\bigvee_{x \in S} x$ and call it the *supremum* of $S$. Similarly, $\bigwedge S = \bigwedge_{x \in S} x$ is the *infimum*. By convention, $\bigwedge \emptyset = 1$ and $\bigvee \emptyset = 0$. For any $a, b \in \mathbf{B}$, we say that $a$ and $b$ are *disjoint* if $a \wedge b = 0$.

**Definition 2** ([17, 21])**.** An element $a \neq 0$ of $\mathbf{B}$ is an *atom* if, for all $x \in \mathbf{B}$, either $x \wedge a = a$ or $x \wedge a = 0$. Equivalently, $a \neq 0$ is an atom if there is no $x \in \mathbf{B}$ such that $0 < x < a$. We say that $\mathbf{B}$ is *atomic* if for every $a \in \mathbf{B} \setminus \{0\}$, there is an atom $x$ such that $x \leq a$.

**Lemma 3** ([12])**.** *For any two distinct atoms $a$, $b \in \mathbf{B}$, $a \wedge b = 0$.*

**Lemma 4** ([14])**.** *The following are equivalent:*

- $\mathbf{B}$ *is atomic.*

[7] We use $+$ to denote symmetric difference because it is the additive operation of a Boolean ring.

- *For any* $x \in \mathbf{B}$, $x = \bigvee_{atoms\ a \leq x} a$.
- $1$ *is the supremum of all atoms.*

**Lemma 5** ([14])**.** *All finite BAs are atomic.*

**Definition 6** ([11, 17])**.** A *measure* on $\mathbf{B}$ is a function $m\colon \mathbf{B} \to \mathbb{R}_{\geq 0}$ such that:

- $m(0) = 0$;
- $m(a \vee b) = m(a) + m(b)$ for all $a, b \in \mathbf{B}$ whenever $a \wedge b = 0$.

If $m(1) = 1$, we call $m$ a *probability measure*. Also, if $m(x) > 0$ for all $x \neq 0$, then $m$ is *strictly positive*.

**Definition 7** ([14])**.** Let $\mathbf{A}$ and $\mathbf{B}$ be BAs. A *(Boolean) homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ is a map $f\colon \mathbf{A} \to \mathbf{B}$ such that:

- $f(x \wedge y) = f(x) \wedge f(y)$,
- $f(x \vee y) = f(x) \vee f(y)$,
- $f(\neg x) = \neg f(x)$

for all $x, y \in \mathbf{A}$.

**Lemma 8** (Homomorphisms preserve order [14])**.** *Let* $f\colon \mathbf{A} \to \mathbf{B}$ *be a homomorphism between two BAs* $\mathbf{A}$ *and* $\mathbf{B}$*. Then, for any* $x, y \in \mathbf{A}$*, if* $x \leq y$*, then* $f(x) \leq f(y)$*.*

**Lemma 9** ([26])**.** *For any* $a, b \in \mathbf{B}$*,* $a \leq b$ *if and only if* $a \wedge \neg b = 0$*.*

**Lemma 10** ([14])**.** *Let* $m\colon \mathbf{B} \to \mathbb{R}_{\geq 0}$ *be a measure. Then for all* $a, b \in \mathbf{B}$*, if* $a \leq b$*, then* $m(a) \leq m(b)$*.*

**Definition 11** ([18])**.** Let $S$ be a set, and let $\mathbf{B}$ be a BA. Then $\mathbf{B}$ is a *free BA over* $S$ if there is a map $S \to \mathbf{B}$ such that for any BA $\mathbf{C}$ and map $S \to \mathbf{C}$, there is a unique homomorphism $\mathbf{B} \to \mathbf{C}$ that makes

$$\begin{tikzcd} S \arrow[r] \arrow[dr] & \mathbf{B} \arrow[d, dashed] \\ & \mathbf{C}. \end{tikzcd}$$

commute. A BA $\mathbf{B}$ is *free* if $S$ exists.

**Lemma 12** ([26])**.** *A finite BA is free if and only if it has* $2^{2^n}$ *elements for some* $n \in \mathbb{N}$*. It then has* $2^n$ *atoms and* $n$ *generators.*

## 3 WMC as a Measure

**Definition 13.** Let $\mathcal{L}$ be a propositional (or first-order) logic, and let $\Delta$ be a theory in $\mathcal{L}$. We can define an equivalence relation on formulas in $\mathcal{L}$ as

$$\alpha \sim \beta \quad \text{if and only if} \quad \Delta \vdash \alpha \leftrightarrow \beta$$

for all $\alpha, \beta \in \mathcal{L}$. Let $[\alpha]$ denote the equivalence class of $\alpha \in \mathcal{L}$ with respect to $\sim$. We can then let $B(\Delta) = \{[\alpha] \mid \alpha \in \mathcal{L}\}$ and define the structure of a BA on $B(\Delta)$ as

$$\begin{aligned} [\alpha] \vee [\beta] &= [\alpha \vee \beta], \\ [\alpha] \wedge [\beta] &= [\alpha \wedge \beta], \\ \neg[\alpha] &= [\neg\alpha], \\ 1 &= [\alpha \to \alpha], \\ 0 &= [\alpha \wedge \neg\alpha] \end{aligned}$$

for all $\alpha, \beta \in \mathcal{L}$. Then $B(\Delta)$ is the *Lindenbaum-Tarski algebra* of $\Delta$ [18, 27].

**Example 14.** Let $\mathcal{L}$ be a propositional logic with $p$ and $q$ as its only atoms. Then $L = \{p, q, \neg p, \neg q\}$ is its set of literals. Let $w : L \to \mathbb{R}_{\geq 0}$ be the *weight function* defined by

$$\begin{aligned} w(p) &= 0.3, \\ w(\neg p) &= 0.7, \\ w(q) &= 0.2, \\ w(\neg q) &= 0.8. \end{aligned}$$

Let $\Delta$ be a theory in $\mathcal{L}$ with a sole axiom $p$. Then $\Delta$ has two models, i.e., $\{p, q\}$ and $\{p, \neg q\}$. The *weighted model count* (WMC) [5] of $\Delta$ is then

$$\sum_{\omega \models \Delta} \prod_{\omega \models l} w(l) = w(p)w(q) + w(p)w(\neg q) = 0.3.$$

The corresponding BA $B(\Delta)$ can then be constructed using Definition 13. Alternatively, one can first construct the free BA generated by the set $\{p, q\}$ and then take a quotient with respect to either the filter generated by $p$ or the ideal[8] generated by $\neg p$.

Each element of $B(\mathcal{L})$ can also be seen as a subset of the set of all models of $\mathcal{L}$, with 0 representing $\emptyset$, 1 representing the set of all (four) models, each atom representing a single model, and each edge going upward representing a subset relation. Thus, the Boolean-algebraic way of calculating the WMC of $\Delta$ consists of:

1. Identifying an element $a \in B(\mathcal{L})$ that corresponds to $\Delta$.
2. Finding all atoms of $B(\mathcal{L})$ that are 'dominated' by $a$ according to the partial order.
3. Using $w$ to calculate the weight of each such atom.
4. Adding the weights of these atoms.

This motivates the following definition of WMC generalised to BAs.

- Why is Step 1 always possible?
- Clarify what $B(L)$ means and whether $B(\Delta)$ is even necessary.
- Find a reference for the set/subset thing.
- This should be replaced with inner sums (a.k.a. free products).
- Mention that the subsequent definition can be reduced to a single formula (i.e., without cases).
- Any measure is a WMC measure if all atoms are in $L$.

**Definition 15.** Let $\mathbf{B}$ be an atomic BA, and let $M \subset \mathbf{B}$ be its set of atoms. Let $L \subset \mathbf{B}$ be such that every atom $m \in M$ can be uniquely expressed as $m = \bigwedge L'$ for some $L' \subseteq L$, and let $w\colon L \to \mathbb{R}_{\geq 0}$ be arbitrary. The *weighted model count* $\mathrm{WMC}_w\colon \mathbf{B} \to \mathbb{R}_{\geq 0}$ is defined as

$$\mathrm{WMC}_w(x) = \begin{cases} 0 & \text{if } x = 0 \\ \prod_{l \in L'} w(l) & \text{if } M \ni x = \bigwedge L' \\ \sum_{\text{atoms } a \leq x} \mathrm{WMC}_w(a) & \text{otherwise} \end{cases}$$

for any $x \in \mathbf{B}$. Furthermore, we define the *normalised weighted model count* $\mathrm{NWMC}_w\colon \mathbf{B} \to [0, 1]$ as $\mathrm{NWMC}_w(x) = \frac{\mathrm{WMC}_w(x)}{\mathrm{WMC}_w(1)}$ for all $x \in \mathbf{B}$. For both $\mathrm{WMC}_w$ and $\mathrm{NWMC}_w$, we will drop the subscript when doing so results in no potential confusion. Finally, we say that a measure $m\colon \mathbf{B} \to \mathbb{R}_{\geq 0}$ is a *WMC measure* (or is *WMC-definable*) if there exists a subset $L \subset \mathbf{B}$ and a weight function $w\colon L \to \mathbb{R}_{\geq 0}$ such that $m = \mathrm{WMC}_w$.

[8] More details on these concepts can be found in many books on BAs [14, 18].

**Theorem 16.** WMC *is a measure, and* NWMC *is a probability measure.*

*Proof.* First, note that WMC is non-negative and $\mathrm{WMC}(0) = 0$ by definition. Next, let $x, y \in \mathbf{B}$ be such that $x \wedge y = 0$. We want to show that

$$\mathrm{WMC}(x \vee y) = \mathrm{WMC}(x) + \mathrm{WMC}(y). \tag{1}$$

If, say, $x = 0$, then Eq. (1) becomes

$$\mathrm{WMC}(y) = \mathrm{WMC}(0) + \mathrm{WMC}(y) = \mathrm{WMC}(y)$$

(and likewise for $y = 0$). Thus we can assume that $x \neq 0 \neq y$ and use Lemma 4 to write

$$x = \bigvee_{i \in I} x_i \quad \text{and} \quad y = \bigvee_{j \in J} y_j$$

for some sequences of atoms $(x_i)_{i \in I}$ and $(y_j)_{j \in J}$. If $x_{i'} = y_{j'}$ for some $i' \in I$ and $j' \in J$, then

$$x \wedge y = \bigvee_{i \in I} \bigvee_{j \in J} x_i \wedge y_j = x_{i'} \wedge y_{j'} \neq 0,$$

contradicting the assumption. This is enough to show that

$$\begin{aligned}\mathrm{WMC}(x \vee y) &= \mathrm{WMC}\left(\left(\bigvee_{i \in I} x_i\right) \vee \left(\bigvee_{j \in J} y_j\right)\right) = \sum_{i \in I} \mathrm{WMC}(x_i) + \sum_{j \in J} \mathrm{WMC}(y_j) \\ &= \mathrm{WMC}(x) + \mathrm{WMC}(y),\end{aligned}$$

finishing the proof that WMC is a measure. This immediately shows that NWMC is a probability measure since, by definition, $\mathrm{NWMC}(1) = 1$. $\square$

Given a theory $\Delta$ in a logic $\mathcal{L}$, the usual way of using WMC to compute the probability of a query $q$ is [1, 25]

$$\Pr_{\Delta, w}(q) = \frac{\mathrm{WMC}_w(\Delta \wedge q)}{\mathrm{WMC}_w(\Delta)}.$$

In our algebraic formulation, this can be computed in two different ways:

- as $\frac{\mathrm{WMC}_w(\Delta \wedge q)}{\mathrm{WMC}_w(\Delta)}$ in $B(\mathcal{L})$,
- and as $\mathrm{NWMC}_w([q])$ in $B(\Delta)$.

But how does the measure defined on $B(\mathcal{L})$ transfer to $B(\Delta)$?

# 4 What Measures Are WMC-Definable?

## 4.1 WMC Requires Independent Literals

**Lemma 17.** *For any measure* $m \colon \mathbf{B} \to \mathbb{R}_{\geq 0}$ *and elements* $a, b \in \mathbf{B}$,

$$m(a \wedge b) = m(a)m(b) \tag{2}$$

*if and only if*

$$m(a \wedge b) \cdot m(\neg a \wedge \neg b) = m(a \wedge \neg b) \cdot m(\neg a \wedge b). \tag{3}$$

*Proof.* First, note that $a = (a \wedge b) \vee (a \wedge \neg b)$ and $(a \wedge b) \wedge (a \wedge \neg b) = 0$, so, by properties of a measure,

$$m(a) = m(a \wedge b) + m(a \wedge \neg b). \tag{4}$$

Applying Eq. (4) and the equivalent expression for $m(b)$ allows us to rewrite Eq. (2) as

$$m(a \wedge b) = [m(a \wedge b) + m(a \wedge \neg b)][m(a \wedge b) + m(\neg a \wedge b)]$$

which is equivalent to

$$m(a \wedge b)[1 - m(a \wedge b) - m(a \wedge \neg b) - m(\neg a \wedge b)] = m(a \wedge \neg b)m(\neg a \wedge b). \tag{5}$$

Since $a \wedge b$, $a \wedge \neg b$, $\neg a \wedge b$, $\neg a \wedge \neg b$ are pairwise disjoint and their supremum is 1,

$$m(a \wedge b) + m(a \wedge \neg b) + m(\neg a \wedge b) + m(\neg a \wedge \neg b) = 1,$$

and this allows us to rewrite Eq. (5) into Eq. (3). As all transformations are invertible, the two expressions are equivalent. ☐

This theorem needs a special case for zero weights.

**Theorem 18.** *Let* $\mathbf{B}$ *be a free BA over* $\{l_i\}_{i=1}^n$ *(for some* $n \in \mathbb{N}$*) with measure* $m\colon \mathbf{B} \to \mathbb{R}_{\geq 0}$*, and let* $L = \{l_i\}_{i=1}^n \cup \{\neg l_i\}_{i=1}^n$*. Then there exists a weight function* $w\colon L \to \mathbb{R}_{\geq 0}$ *such that* $m = \mathrm{WMC}_w$ *if and only if*

$$m(l \wedge l') = m(l)m(l') \tag{6}$$

*for all distinct* $l, l' \in L$ *such that* $l \neq \neg l'$.

*Remark.* Note that if $n = 1$, then Eq. (6) is vacuously satisfied and so any valid measure can be expressed as WMC.

*Proof.* ($\Leftarrow$) Let $w\colon L \to \mathbb{R}_{\geq 0}$ be defined by

$$w(l) = m(l) \tag{7}$$

for all $l \in L$. We are going to show that $\mathrm{WMC}_w = m$. First, note that $\mathrm{WMC}_w(0) = 0 = m(0)$ by the definitions of both $\mathrm{WMC}_w$ and $m$. Second, let

$$a = \bigwedge_{i=1}^n a_i \tag{8}$$

be an atom in $\mathbf{B}$ such that $a_i \in \{l_i, \neg l_i\}$ for all $i \in [n]$. Then

$$\mathrm{WMC}(a) = \prod_{i=1}^n w(a_i) = \prod_{i=1}^n m(a_i) = m\left(\bigwedge_{i=1}^n a_i\right) = m(a)$$

by Definition 15 and Eqs. (6) to (8). Finally, note that if WMC and $m$ agree on all atoms, then they must also agree on all other non-zero elements of the Boolean algebra.

($\Rightarrow$) For the other direction, we are given a weight function $w\colon L \to \mathbb{R}_{\geq 0}$ that induces a measure $m = \mathrm{WMC}_w\colon \mathbf{B} \to \mathbb{R}_{\geq 0}$, and we want to show that Eq. (6) is satisfied. Let $k_i, k_j \in L$ be such that $k_i \in \{l_i, \neg l_i\}$, $k_j \in \{l_j, \neg l_j\}$, and $i \neq j$ for some $i, j \in [n]$. We then want to show that

$$m(k_i \wedge k_j) = m(k_i)m(k_j) \tag{9}$$

which is equivalent to

$$m(k_i \wedge k_j) \cdot m(\neg k_i \wedge \neg k_j) = m(k_i \wedge \neg k_j) \cdot m(\neg k_i \wedge k_j) \tag{10}$$

by Lemma 17. Then

$$\begin{aligned}
\mathrm{WMC}(k_i \wedge k_j) &= \sum_{\text{atoms } a \leq k_i \wedge k_j} \mathrm{WMC}(a) = \sum_{\text{atoms } a \leq k_i \wedge k_j} \prod_{m \in [n]} w(a_m) \\
&= \sum_{\text{atoms } a \leq k_i \wedge k_j} w(a_i) w(a_j) \prod_{m \in [n] \setminus \{i,j\}} w(a_m) = \sum_{\text{atoms } a \leq k_i \wedge k_j} w(k_i) w(k_j) \prod_{m \in [n] \setminus \{i,j\}} w(a_m) \\
&= w(k_i) w(k_j) \sum_{\text{atoms } a \leq k_i \wedge k_j} \prod_{m \in [n] \setminus \{i,j\}} w(a_m) = w(k_i) w(k_j) C,
\end{aligned}$$

where $C$ denotes the part of $\mathrm{WMC}(k_i \wedge k_j)$ that will be the same for $\mathrm{WMC}(\neg k_i \wedge k_j)$, $\mathrm{WMC}(k_i \wedge \neg k_j)$, and $\mathrm{WMC}(\neg k_i \wedge \neg k_j)$ as well. But then Eq. (10) becomes

$$w(k_i) w(k_j) w(\neg k_i) w(\neg k_j) C^2 = w(k_i) w(\neg k_j) w(\neg k_i) w(k_j) C^2$$

which is trivially true. □

## 4.2 Extending the Algebra

Given this requirement for independence, a well-known way to represent probability distributions that do not consist entirely of independent variables is by adding more literals [5], i.e., extending the set $L$ covered by the WMC weight function $w \colon L \to \mathbb{R}_{\geq 0}$. Let us translate this idea to the language of BAs.

**Theorem 19.** *Let $\mathbf{B}$ be a free BA over a finite set $S$, and let $m \colon \mathbf{B} \to \mathbb{R}_{\geq 0}$ be an arbitrary measure. Let $L = \{s \mid s \in S\} \cup \{\neg s \mid s \in S\}$. By Lemma 12, we know that $\mathbf{B}$ has $n = 2^{|S|}$ atoms. Let $\{a_i\}_{i=1}^n$ denote those atoms in some arbitrary order. Let $L' = L \cup \{\phi_i\}_{i=1}^n \cup \{\neg \phi_i\}_{i=1}^n$ be the set $L$ extended with $2n$ new elements. Let $\mathbf{B}'$ be the unique Boolean algebra with $\{\phi_i \wedge a_i\}_{i=1}^n \cup \{\neg \phi_i \wedge a_i\}_{i=1}^n$ as its set of atoms. Let $\iota \colon \mathbf{B} \hookrightarrow \mathbf{B}'$ be the inclusion homomorphism. Let $w \colon L' \to \mathbb{R}_{\geq 0}$ be defined by*

$$w(l) = \begin{cases} m(a_i)/2 & \textit{if } l = \phi_i \textit{ or } l = \neg \phi_i \textit{ for some } i \in [n] \\ 1 & \textit{otherwise} \end{cases}$$

*for all $l \in L'$, and note that this defines a WMC measure $m' = \mathrm{WMC}_w \colon \mathbf{B}' \to \mathbb{R}_{\geq 0}$. Then*

$$m(a) = (m' \circ \iota)(a)$$

*for all $a \in \mathbf{B}$.*

In other words, any measure can be computed using WMC by extending the BA with more literals. More precisely, we are given the left-hand column in

$$\begin{tikzcd}
\mathbb{R}_{\geq 0} & & \\
\mathbf{B} \arrow[u, "m"] \arrow[r, hook, "\iota"] & \mathbf{B}' \arrow[ul, "m'"'] & \\
L \arrow[u, hook] \arrow[r, hook] & L' \arrow[u, hook] \arrow[r, "w"] & \mathbb{R}_{\geq 0}
\end{tikzcd}$$

and construct the remaining part in such a way that the triangle commutes.

- Make $J$ depend on $i$.
- Find a reference for this first claim in the following proof.

*Proof.* Since $\mathbf{B}$ is free over $S$, each atom $a_i \in \mathbf{B}$ is an infimum of elements in $L$, i.e.,

$$a_i = \bigwedge_{j \in J} a_{i,j}$$

for some $\{a_{i,j}\}_{j \in J} \subset L$. Moreover, each atom $b \in \mathbf{B}'$ can be represented as either $b = \phi_i \wedge a_i$ or $b = \neg\phi_i \wedge a_i$ for some atom $a_i \in \mathbf{B}$, also making it an infimum over a subset of $L'$. Then, for any $b \in \mathbf{B}$,

$$(m' \circ \iota)(b) = \sum_{\substack{\text{atoms } a_i \in \mathbf{B}: \\ \phi_i \wedge a_i \leq \iota(b)}} (w(\phi_i) + w(\neg\phi_i)) \prod_{j \in J} w(a_{i,j}),$$

recognising that, for any $\iota(b)$, any atom $a_i \in \mathbf{B}$ satisfies $\phi_i \wedge a_i \leq \iota(b)$ if and only if it satisfies $\neg\phi_i \wedge a_i \leq \iota(b)$. Then, according to the definition of $w$,

$$(m' \circ \iota)(b) = \sum_{\substack{\text{atoms } a_i \in \mathbf{B}: \\ \phi_i \wedge a_i \leq \iota(b)}} (w(\phi_i) + w(\neg\phi_i)) = \sum_{\substack{\text{atoms } a_i \in \mathbf{B}: \\ \phi_i \wedge a_i \leq \iota(b)}} m(a_i) = m(b),$$

provided that

$$\phi_i \wedge a_i \leq \iota(b) \quad \text{if and only if} \quad a_i \leq b,$$

but this is equivalent to

$$\phi_i \wedge a_i = \phi_i \wedge a_i \wedge b \quad \text{if and only if} \quad a_i = a_i \wedge b$$

which is true because $\phi_i \notin L$. $\square$

Now we can show that the construction in Theorem 19 is smallest possible.

**Conjecture 20.** *Let $\mathbf{B}$ and $\mathbf{B}'$ be Boolean algebras, and $\iota \colon \mathbf{B} \hookrightarrow \mathbf{B}'$ be the inclusion map such that $\mathbf{B}$ is free over $L$, all atoms of $\mathbf{B}'$ can be expressed as meets of elements of $L'$, and the following subset relations are satisfied:*

$$\begin{array}{ccc} \mathbf{B} & \xhookrightarrow{\iota} & \mathbf{B}' \\ \cup & & \cup \\ L & \subset & L' \end{array}$$

*If, for any measure $m \colon \mathbf{B} \to \mathbb{R}_{\geq 0}$, one can construct a weight function $w \colon L' \to \mathbb{R}_{\geq 0}$ such that the WMC measure* $\mathrm{WMC} \colon \mathbf{B}' \to \mathbb{R}_{\geq 0}$ *with respect to $w$ satisfies*

$$m = \mathrm{WMC} \circ \iota,$$

*then $|L' \setminus L| \geq 2^{|L|+1}$.*

Let us note how our lower bound on the number of added literals compares to two methods of translating a discrete probability distribution into a WMC problem over a propositional knowledge base proposed by Darwiche [7] and Sang et al. [25]. Suppose we have a discrete probability distribution with $n$ variables, and the $i$th variable has $v_i$ values, for each $i \in [n]$. Interpreted as a logical system, it has $\prod_{i=1}^{n} v_i$ models. My expansion would then use

$$\sum_{i=1}^{n} v_i + 2 \prod_{i=1}^{n} v_i$$

variables, i.e., a variable for each possible variable-value assignment, and two additional variables for each model. Without making any independence assumptions, the encoding by Darwiche [7] would use

$$\sum_{i=1}^{n} v_i + \sum_{i=1}^{n} \prod_{j=1}^{i} v_j$$

variables, while for the encoding by Sang et al. [25],

$$\sum_{i=1}^{n} v_i + \sum_{i=1}^{n} (v_i - 1) \prod_{j=1}^{i-1} v_j$$

variables would suffice.

# 5 Representing Independence and Conditional Independence

## 5.1 Preliminaries

**Definition 21.** Given a BA $\mathbf{A}$, a *subalgebra* is a subset $\mathbf{B} \subseteq \mathbf{A}$ that, together with the operations, zero, and one of $\mathbf{A}$, is a BA.

**Definition 22** ([14])**.** Let $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ be BAs such that $\mathbf{B}$ is a subalgebra of $\mathbf{A}$. Let $f\colon \mathbf{A} \to \mathbf{C}$ and $g\colon \mathbf{B} \to \mathbf{C}$ be homomorphisms. Then $f$ is an *extension* of $g$ if $f(x) = g(x)$ for all $x \in \mathbf{B}$. If $f$ is an extension of each member of a family $\{g_i\}_{i \in I}$ of homomorphisms, then $f$ is called a *common extension* of $\{g_i\}_{i \in I}$.

**Definition 23** ([14])**.** Let $\{\mathbf{A}_i\}_{i \in I}$ be a family of subalgebras of a BA $\mathbf{A}$. If for any BA $\mathbf{B}$ with a family of homomorphisms $\{f_i\colon \mathbf{A}_i \to \mathbf{B}\}_{i \in I}$ there exists a unique common extension of $\{f_i\colon \mathbf{A}_i \to \mathbf{B}\}_{i \in I}$ ($f\colon \mathbf{A} \to \mathbf{B}$ in the diagram),

$$\begin{tikzcd} \mathbf{A}_i \arrow[r, hook] \arrow[dr, "f_i"'] & \mathbf{A} \arrow[d, dashed, "f"] \\ & \mathbf{B} \end{tikzcd}$$

then $\mathbf{A}$ is the *internal sum*[9] of $\{\mathbf{A}_i\}_{i \in I}$. We will denote it as $\bigoplus_{i \in I} \mathbf{A}_i$.

**Proposition 24** ([26])**.** *Let $\mathbf{A}$ be the internal sum of a family of BAs $\{\mathbf{A}_i\}_{i \in I}$, and let $\{m_i\colon \mathbf{A}_i \to \mathbb{R}_{\geq 0}\}_{i \in I}$ be a family of measures. Then there exists a unique measure $m\colon \mathbf{A} \to \mathbb{R}_{\geq 0}$ such that, for any finite subset $J \subseteq I$ and family of elements $\{x_j \in \mathbf{A}_j\}_{j \in J}$,*

$$m\left(\bigwedge_{j \in J} x_j\right) = \prod_{j \in J} m_j(x_j).$$

**Definition 25** ([18])**.** Let $\mathbf{A}$ be a BA. Let $\mathbf{B}$ be a subalgebra of $\mathbf{A}$, and let $\{\mathbf{A}_i\}_{i \in I}$ be a family of subalgebras of $\mathbf{A}$ such that $\mathbf{A}_i \cap \mathbf{A}_j = \mathbf{B}$ for all $i \neq j$ in $I$. Let $\{\iota_i\colon \mathbf{B} \to \mathbf{A}_i\}$ be a family of inclusion homomorphisms. Then $\mathbf{A}$ is the *amalgamated free product*[10] *of* $\{\mathbf{A}_i\}_{i \in I}$ *over* $\mathbf{B}$ if, for any Boolean algebra $\mathbf{C}$ with a family of homomorphisms $\{f_i\colon \mathbf{A}_i \to \mathbf{C}\}_{i \in I}$ such that $f_i \circ \iota_i = f_j \circ \iota_j$ for all $i, j \in I$, there is a unique homomorphism $f\colon \mathbf{A} \to \mathbf{C}$ such that the triangle in

$$\begin{tikzcd} \mathbf{B} \arrow[r, hook, "\iota_i"] & \mathbf{A}_i \arrow[r, hook] \arrow[d, "f_i"'] & \mathbf{A} \arrow[dl, dashed, "f"] \\ & \mathbf{C} & \end{tikzcd}$$

commutes for all $i \in I$. We will denote this product as

$$\mathbf{A} = \bigoplus_{\substack{\mathbf{B} \\ i \in I}} \mathbf{A}_i.$$

[9] A slightly more general version of this definition is also known as the free product, the Boolean product, and the coproduct in the category of BAs [14, 18, 26].

[10] Also known as a (wide) pushout in the category of BAs.

## 5.2 New Results

Make sure that this is enough to guarantee a pushout.

**Theorem 26.** *Let $\{S_i\}_{i=0}^n$ be a finite set of finite sets for some $n > 1$ such that for all distinct positive integers $i$ and $j$, $S_i \cap S_j = S_0$, and let*

$$\mathbf{A} = \bigoplus_{\substack{\mathcal{F}(S_0)\\ 1 \le i \le n}} \mathcal{F}(S_i).$$

*Let $(m_i \colon \mathcal{F}(S_i) \to \mathbb{R}_{\ge 0})_{i=1}^n$ be arbitrary measures. Then there is a unique measure $m \colon \mathbf{A} \to \mathbb{R}_{\ge 0}$ such that, for any element $b \in \mathcal{F}(S_0)$, subset $J \subseteq \{1, 2, \dots, n\}$, and elements $\{a_j \in \mathcal{F}(S_j \setminus S_0)\}_{j \in J}$,*

$$m\left(b \wedge \bigwedge_{j \in J} a_j\right) = \prod_{j \in J} m_j(b \wedge a_j).$$

**Theorem 27.** *The number of weights needed to encode a Bayesian network using coproducts and pushouts is equal to the number of entries in the tables of the network (and the resulting theory is shorter).*

**Theorem 28** (Pushouts of free BAs are free)**.** *Let*

$$\mathbf{A} = \bigoplus_{\substack{\mathbf{B}\\ i \in I}} \mathbf{A}_i$$

*be an amalgamated free product such that $\{\mathbf{A}_i\}_{i \in I}$ are free BAs with $\{S_i\}_{i \in I}$ as their respective sets of generators. Let $S = \bigcup_{i \in I} S_i$. Then $\mathbf{A}$ is a free BA with generating set $S$.*

*Proof.* Suppose we have a map from $S$ to an arbitrary BA $\mathbf{C}$, as in

$$\begin{tikzcd}
S_i \arrow[r, hook] \arrow[d, hook] & S \arrow[d, hook] \arrow[ddr, bend left] & \\
\mathbf{A}_i \arrow[r, hook] \arrow[drr, dashed, bend right] & \mathbf{A} \arrow[dr, dashed] & \\
& & \mathbf{C}.
\end{tikzcd}$$

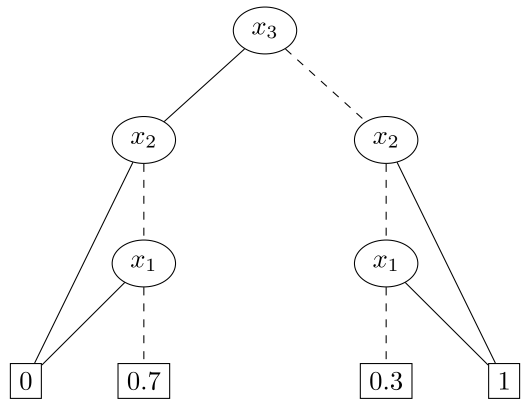
We want to show that there exists a unique homomorphism $\mathbf{A} \to \mathbf{C}$. For all $i \in I$, from $S_i \hookrightarrow S$ and $S \to \mathbf{C}$ we get a map $S_i \to \mathbf{C}$, so—by the definition of a free BA—there is a unique homomorphism $\mathbf{A}_i \to \mathbf{C}$. Furthermore, a family of homomorphisms $\{\mathbf{A}_i \to \mathbf{C}\}_{i \in I}$ uniquely determine a homomorphism $\mathbf{A} \to \mathbf{C}$ by the universal mapping property of a (wide) pushout. Thus $\mathbf{A}$ is a free BA with generating set $S$. ☐

**Corollary 29.** *Similarly, coproducts of free BAs are free.*

# 6 Representation of conditional probability tables

The probability of a variable with $n$ values $x_1, \dots, x_n$ (whether conditioned on something or not) can be represented with an ADD of size $\Theta(n^2)$ by representing each of $\Pr(x_1), \Pr(x_2 \mid x_1), \dots, \Pr(x_n \mid x_1, \dots, x_{n-1})$

by a separate ADD. For example, if $n = 3$, the last of the three ADDs would look like this




Such an ADD for $x_n$ has $2n+3$ nodes and $4n-2$ edges. The same information can be represented in textual form as

```
w 3 1 0
w 3 2 0
w 3 -1 -2 0.7
```

The encoding weights depend on the probabilities in the CPT as follows. For each value $x_n$, if $\Pr(X = x_n) = 0$, then $w(x_n) = 0$. Otherwise, let $I_n = \{1 \le i \le n-1 \mid \Pr(X = x_i) \neq 0\}$. Then

$$w\left(x_n \,\middle|\, \bigwedge_{i \in I_n} \neg x_i\right) = \Pr(X = x_n) \prod_{i \in I_n} \frac{1}{1 - w\left(x_i \,\middle|\, \bigwedge_{j \in I_i} \neg x_j\right)}.$$

All of these weights/probabilities may have other conditions that come from the structure of the Bayesian network.

# 7 Encoding Bayesian networks

Let $V$ denote the set of random variables in a Bayesian network. For any random variable $X \in V$, let $\mathrm{pa}(X)$ denote the set of parents of $X$ and $\operatorname{im} X$ denote the set of possible values.

**Definition 30** (Indicator variables)**.** Let $X \in V$ be a random variable. If $X$ is binary (i.e., $|\operatorname{im} X| = 2$), we can arbitrary identify one of the values as 1 and the other one as 0 (i.e, $\operatorname{im} X \cong \{0, 1\}$). Then $X$ can be represented by a single *indicator variable* $\lambda_{X=1}$. If we interpret $2^{\{\lambda_{X=1}\}}$ as a Boolean algebra, we can let $\lambda_{X=0} = \neg\lambda_{X=1}$ in the algebraic notation, or, equivalently, $\lambda_{X=0} = \emptyset \in 2^{\{\lambda_{X=1}\}}$ in the set-theoretic notation. On the other hand, if $X$ is not binary, we represent $X$ with $|\operatorname{im} X|$ indicator variables, one for each value. We let

$$E(X) = \begin{cases} \{\lambda_{X=1}\} & \text{if } |\operatorname{im} X| = 2 \\ \{\lambda_{X=x} \mid x \in \operatorname{im} X\} & \text{otherwise.} \end{cases}$$

denote the set of indicator variables for $X$ and

$$E^*(X) = E(X) \cup \bigcup_{Y \in \mathrm{pa}(X)} E(Y).$$

denote the set of indicator variables for $X$ and its parents in the Bayesian network. Finally, let

$$U = \bigcup_{X \in V} E(X)$$

denote the set of all indicator variables for all random variables in the Bayesian network.

**Definition 31** (Operations on functions)**.** Let $A\colon 2^X \to \mathbb{R}_{\geq 0}$ and $B\colon 2^Y \to \mathbb{R}_{\geq 0}$ be Boolean functions, $\alpha \in \mathbb{R}_{\geq 0}$, and $x \in X$. We define the following operations:

**Addition:** $A + B$ is a function $A + B\colon 2^{X \cup Y} \to \mathbb{R}_{\geq 0}$ such that

$$(A + B)(\tau) = A(\tau \cap X) + B(\tau \cap Y)$$

for all $\tau \in 2^{X \cup Y}$.

**Inverse:** $\overline{A}$ is a function $\overline{A}\colon 2^X \to \mathbb{R}_{\geq 0}$ such that

$$\overline{A}(\tau) = 1 - A(\tau)$$

for all $\tau \in 2^X$.

**Multiplication:** $A \cdot B$ is a function $A \cdot B\colon 2^{X \cup Y} \to \mathbb{R}_{\geq 0}$ such that

$$(A \cdot B)(\tau) = A(\tau \cap X) \cdot B(\tau \cap Y)$$

for all $\tau \in 2^{X \cup Y}$.

**Scalar multiplication:** $\alpha A$ is a function $\alpha A\colon 2^X \to \mathbb{R}_{\geq 0}$ such that

$$(\alpha A)(\tau) = \alpha \cdot A(\tau)$$

for all $\tau \in 2^X$.

**Projection:** $\exists_x A$ is a function $\exists_x A\colon 2^{X \setminus \{x\}} \to \mathbb{R}_{\geq 0}$ such that

$$(\exists_x A)(\tau) = A(\tau) + A(\tau \cup \{x\})$$

for all $\tau \in 2^{X \setminus \{x\}}$.

Note that both addition and multiplication commute.

**Definition 32** (Special functions)**.**

- unit $1\colon 2^\emptyset \to \mathbb{R}_{\geq 0}$, $1(\tau) = 1$.
- zero $0\colon 2^\emptyset \to \mathbb{R}_{\geq 0}$, $0(\tau) = 0$.
- constant $[a]\colon 2^{\{a\}} \to \mathbb{R}_{\geq 0}$,

$$[a](\tau) = \begin{cases} 1 & \text{if } a \in \tau \\ 0 & \text{if } a \notin \tau. \end{cases}$$

Henceforth, for any function $A\colon 2^X \to \mathbb{R}_{\geq 0}$ and any set $\tau$, we will write $A(\tau)$ to mean $A(\tau \cap X)$.

**Lemma 33.** *Let $X \in V$ be a random variable with parents $\mathrm{pa}(X) = \{Y_1, \dots, Y_n\}$. Then $\mathrm{CPT}_X\colon 2^{E^*(X)} \to \mathbb{R}_{\geq 0}$ is such that for any $x \in \operatorname{im} X$ and $(y_1, \dots, y_n) \in \prod_{i=1}^n \operatorname{im} Y_i$,*

$$\mathrm{CPT}_X\left(\lambda_{X=x} \wedge \bigwedge_{i=1}^{n} \lambda_{Y_i=y_i}\right) = \Pr(X = x \mid Y_1 = y_1, \dots, Y_n = y_n).$$

```
φ ← 1;
for X ∈ V do
    let pa(X) = {Y_1, ..., Y_n};
    CPT_X ← 0;
    if |im X| = 2 then
        for (y_1, ..., y_n) ∈ ∏_{i=1}^n im Y_i do
            p_1 ← Pr(X = 1 | Y_1 = y_1, ..., Y_n = y_n);
            p_0 ← Pr(X ≠ 1 | Y_1 = y_1, ..., Y_n = y_n);
            CPT_X ← CPT_X + p_1[λ_{X=1}] · ∏_{i=1}^n [λ_{Y_i=y_i}] + p_0 overline([λ_{X=1}]) · ∏_{i=1}^n [λ_{Y_i=y_i}];
    else
        let im X = {x_1, ..., x_m};
        for x ∈ im X and (y_1, ..., y_n) ∈ ∏_{i=1}^n im Y_i do
            p_x ← Pr(X = x | Y_1 = y_1, ..., Y_n = y_n);
            CPT_X ← CPT_X + p_x[λ_{X=x}] · ∏_{i=1}^n [λ_{Y_i=y_i}] + overline([λ_{X=1}]) · ∏_{i=1}^n [λ_{Y_i=y_i}];
        CPT_X ← CPT_X · (∑_{i=1}^m [λ_{X=x_i}]) · ∏_{i=1}^m ∏_{j=i+1}^m (overline([λ_{X=x_i}]) + overline([λ_{X=x_j}]));
    φ ← φ · CPT_X;
return φ;
```

*Proof.* Let

$$\tau = \lambda_{X=x} \wedge \bigwedge_{i=1}^{n} \lambda_{Y_i=y_i}.$$

If $X$ is binary, then $\mathrm{CPT}_X$ is a sum of $2\prod_{i=1}^n |\operatorname{im} Y_i|$ terms, one for each possible assignment of values to variables $X, Y_1, \ldots, Y_n$. Exactly one of these terms is nonzero when applied to $\tau$, and it is equal to $\Pr(X = x \mid Y_1 = y_1, \ldots, Y_n = y_n)$ by definition.

If $X$ is not binary, then

$$\left(\sum_{i=1}^{m} [\lambda_{X=x_i}]\right)(\tau) = 1,$$

and

$$\left(\prod_{i=1}^{m} \prod_{j=i+1}^{m} (\overline{[\lambda_{X=x_i}]} + \overline{[\lambda_{X=x_j}]})\right)(\tau) = 1,$$

so, by a similar argument as before,

$$\mathrm{CPT}_X(\tau) = \Pr(X = x \mid Y_1 = y_1, \ldots, Y_n = y_n).$$

□

**Proposition 34.** *$\phi \colon 2^U \to \mathbb{R}_{\geq 0}$ represents the full probability distribution of the Bayesian network, i.e., if $V = \{X_1, \ldots, X_n\}$, then*

$$\phi(\tau) = \begin{cases} \Pr(X_1 = x_1, \ldots, X_n = x_n) & \text{if } \tau = \bigwedge_{i=1}^n \lambda_{X_i=x_i} \text{ for some } (x_1, \ldots, x_n) \in \prod_{i=1}^n \operatorname{im} X_i \\ 0 & \text{otherwise,} \end{cases}$$

*for all $\tau \in 2^U$.*

*Proof.* If

$$\tau = \bigwedge_{X \in V} \lambda_{X=v_X}$$

for some $(v_X)_{X \in V} \in \prod_{X \in V} \operatorname{im} X$, then

$$\phi(\tau) = \prod_{X \in V} \Pr\left(X = v_X \;\middle|\; \bigwedge_{Y \in \mathrm{pa}(X)} Y = v_Y\right) = \Pr\left(\bigwedge_{X \in V} X = v_X\right)$$

by Lemma 33 and the definition of a Bayesian network. Otherwise there must be some non-binary random variable $X \in V$ such that $|E(X) \cap \tau| \neq 1$. If $E(X) \cap \tau = \emptyset$, then

$$\left(\sum_{i=1}^{m} [\lambda_{X=x_i}]\right)(\tau) = 0,$$

and so $\mathrm{CPT}_X(\tau) = 0$, and $\phi(\tau) = 0$. If $|E(X) \cap \tau| > 1$, then we must have two different values $x_1, x_2 \in \operatorname{im} X$ such that $\{\lambda_{X=x_1}, \lambda_{X=x_2}\} \subseteq \tau$ which means that

$$(\overline{[\lambda_{X=x_1}]} + \overline{[\lambda_{X=x_2}]})(\tau) = 0,$$

and so, again, $\mathrm{CPT}_X(\tau) = 0$, and $\phi(\tau) = 0$. □

**Theorem 35.** *Let $\phi \colon 2^U \to \mathbb{R}_{\geq 0}$ be a function generated by the algorithm. Then*

$$(\exists_U (\phi \cdot [\lambda_{X=x}]))(\emptyset) = \Pr(X = x).$$

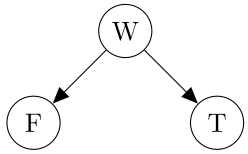
*Proof.* Let $V = \{X, Y_1, \dots, Y_n\}$. Then

$$\begin{aligned}(\exists_U (\phi \cdot [\lambda_{X=x}]))(\emptyset) &= \sum_{\tau \in 2^U} (\phi \cdot [\lambda_{X=x}])(\tau) = \sum_{\lambda_{X=x} \in \tau \in 2^U} \phi(\tau) = \sum_{\lambda_{X=x} \in \tau \in 2^U} \left(\prod_{Y \in V} \mathrm{CPT}_Y\right)(\tau) \\ &= \sum_{(y_1, \dots, y_n) \in \prod_{i=1}^{n} \operatorname{im} Y_i} \Pr(X = x, Y_1 = y_1, \dots, Y_n = y_n) = \Pr(X = x)\end{aligned}$$

by the following arguments:

- the proof of Theorem 1 in the ADDMC paper [9];
- if $\lambda_{X=x} \notin \tau \in 2^U$, then $(\phi \cdot [\lambda_{X=x}])(\tau) = \phi(\tau) \cdot [\lambda_{X=x}](\tau \cap \{\lambda_{X=x}\}) = \phi(\tau) \cdot 0 = 0$;
- Proposition 34;
- marginalisation of a probability distribution.

□



| $w$ | $\Pr(W = w)$ |
|---|---|
| 1 | 0.5 |
| 0 | 0.5 |

| $w$ | $f$ | $\Pr(F = f \mid W = w)$ |
|---|---|---|
| 1 | 1 | 0.6 |
| 1 | 0 | 0.4 |
| 0 | 1 | 0.1 |
| 0 | 0 | 0.9 |

| $w$ | $t$ | $\Pr(T = t \mid W = w)$ |
|---|---|---|
| 1 | $l$ | 0.2 |
| 1 | $m$ | 0.4 |
| 1 | $h$ | 0.4 |
| 0 | $l$ | 0.6 |
| 0 | $m$ | 0.3 |
| 0 | $h$ | 0.1 |

Figure 1: An example Bayesian network with its CPTs

**Example 36.** The Bayesian network in Fig. 1 has

$$\begin{aligned}
V &= \{W, F, T\},\\
\mathrm{pa}(W) &= \emptyset,\\
\mathrm{pa}(F) &= \mathrm{pa}(T) = \{W\},\\
\operatorname{im} W &= \operatorname{im} F = \{0, 1\},\\
\operatorname{im} T &= \{l, m, h\},\\
E(W) &= \{\lambda_{W=1}\},\\
E(F) &= \{\lambda_{F=1}\},\\
E(T) &= \{\lambda_{T=l}, \lambda_{T=m}, \lambda_{T=h}\},\\
E^*(W) &= \{\lambda_{W=1}\},\\
E^*(F) &= \{\lambda_{F=1}, \lambda_{W=1}\},\\
E^*(T) &= \{\lambda_{T=l}, \lambda_{T=m}, \lambda_{T=h}, \lambda_{W=1}\},\\
\mathrm{CPT_W} &= 0.5[\lambda_{W=1}] + 0.5\overline{[\lambda_{W=1}]} = 0.5 \cdot 1,\\
\mathrm{CPT_F} &= 0.6[\lambda_{F=1}] \cdot [\lambda_{W=1}] + 0.4[\lambda_{F=0}] \cdot [\lambda_{W=1}] + 0.1[\lambda_{F=1}] \cdot [\lambda_{W=0}] + 0.9[\lambda_{F=0}] \cdot [\lambda_{W=0}]\\
&= 0.6[\lambda_{F=1}] \cdot [\lambda_{W=1}] + 0.4\overline{[\lambda_{F=1}]} \cdot [\lambda_{W=1}] + 0.1[\lambda_{F=1}] \cdot \overline{[\lambda_{W=1}]} + 0.9\overline{[\lambda_{F=1}]} \cdot \overline{[\lambda_{W=1}]},\\
\mathrm{CPT_T} &= ([\lambda_{T=l}] + [\lambda_{T=m}] + [\lambda_{T=h}]) \cdot (\overline{[\lambda_{T=l}]} + \overline{[\lambda_{T=m}]}) \cdot (\overline{[\lambda_{T=l}]} + \overline{[\lambda_{T=h}]}) \cdot (\overline{[\lambda_{T=m}]} + \overline{[\lambda_{T=h}]}) \cdot (\dots),
\end{aligned}$$

and can be encoded in a DIMACS-like CNF format as

| | | | | |
|---|---|---|---|---|
| $\lambda_{T=l}$ | $\lambda_{T=m}$ | $\lambda_{T=h}$ | | 0 |
| | $-\lambda_{T=l}$ | $-\lambda_{T=m}$ | | 0 |
| | $-\lambda_{T=l}$ | $-\lambda_{T=h}$ | | 0 |
| | $-\lambda_{T=m}$ | $-\lambda_{T=h}$ | | 0 |
| $w$ | $\lambda_{W=1}$ | | 0.5 | 0.5 |
| $w$ | $\lambda_{F=1}$ | $\lambda_{W=1}$ | 0.6 | 0.4 |
| $w$ | $\lambda_{F=1}$ | $-\lambda_{W=1}$ | 0.1 | 0.9 |
| $w$ | $\lambda_{T=l}$ | $\lambda_{W=1}$ | 0.2 | 1 |
| $w$ | $\lambda_{T=m}$ | $\lambda_{W=1}$ | 0.4 | 1 |
| $w$ | $\lambda_{T=h}$ | $\lambda_{W=1}$ | 0.4 | 1 |
| $w$ | $\lambda_{T=l}$ | $\lambda_{W=0}$ | 0.6 | 1 |
| $w$ | $\lambda_{T=m}$ | $\lambda_{W=0}$ | 0.3 | 1 |
| $w$ | $\lambda_{T=h}$ | $\lambda_{W=0}$ | 0.1 | 1 |

with each $\lambda$ replaced with a unique positive integer.

# References


[1] Vaishak Belle. Weighted model counting with function symbols. In Gal Elidan, Kristian Kersting, and Alexander T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017.

[2] Elena Castiñeira, Susana Cubillo, and Enric Trillas. On possibility and probability measures in finite Boolean algebras. *Soft Comput.*, 7(2):89–96, 2002.

[3] Mark Chavira and Adnan Darwiche. Compiling Bayesian networks with local structure. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 1306–1312. Professional Book Center, 2005.

[4] Mark Chavira and Adnan Darwiche. Encoding CNFs to empower component analysis. In Armin Biere and Carla P. Gomes, editors, *Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings*, volume 4121 of *Lecture Notes in Computer Science*, pages 61–74. Springer, 2006.

[5] Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artif. Intell.*, 172(6-7):772–799, 2008.

[6] Mark Chavira, Adnan Darwiche, and Manfred Jaeger. Compiling relational Bayesian networks for exact inference. *Int. J. Approx. Reason.*, 42(1-2):4–20, 2006.

[7] Adnan Darwiche. A logical approach to factoring belief networks. In Dieter Fensel, Fausto Giunchiglia, Deborah L. McGuinness, and Mary-Anne Williams, editors, *Proceedings of the Eights International Conference on Principles and Knowledge Representation and Reasoning (KR-02), Toulouse, France, April 22-25, 2002*, pages 409–420. Morgan Kaufmann, 2002.

[8] Adnan Darwiche. New advances in compiling CNF into decomposable negation normal form. In Ramón López de Mántaras and Lorenza Saitta, editors, *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004*, pages 328–332. IOS Press, 2004.

[9] Jeffrey M. Dudek, Vu Phan, and Moshe Y. Vardi. ADDMC: weighted model counting with algebraic decision diagrams. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 1468–1476. AAAI Press, 2020.

[10] Daan Fierens, Guy Van den Broeck, Ingo Thon, Bernd Gutmann, and Luc De Raedt. Inference in probabilistic logic programs using weighted CNF's. In Fábio Gagliardi Cozman and Avi Pfeffer, editors, *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011*, pages 211–220. AUAI Press, 2011.

[11] Haim Gaifman. Concerning measures on Boolean algebras. *Pacific Journal of Mathematics*, 14(1):61–73, 1964.

[12] M. Ganesh. *Introduction to fuzzy sets and fuzzy logic.* PHI Learning Pvt. Ltd., 2006.

[13] Scott Garrabrant, Tsvi Benson-Tilsen, Andrew Critch, Nate Soares, and Jessica Taylor. Logical induction. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:154, 2016.

[14] Steven Givant and Paul R. Halmos. *Introduction to Boolean algebras.* Springer Science & Business Media, 2008.

[15] Theodore Hailperin. Probability logic. *Notre Dame Journal of Formal Logic*, 25(3):198–212, 1984.

[16] Steven Holtzen, Guy Van den Broeck, and Todd D. Millstein. Dice: Compiling discrete probabilistic programs for scalable inference. *CoRR*, abs/2005.09089, 2020.

[17] Thomas Jech. *Set theory, Second Edition*. Perspectives in Mathematical Logic. Springer, 1997.

[18] Sabine Koppelberg, Robert Bonnet, and James Donald Monk. *Handbook of Boolean algebras*, volume 384. North-Holland Amsterdam, 1989.

[19] Peter H. Krauss. Representation of conditional probability measures on Boolean algebras. *Acta Mathematica Hungarica*, 19(3-4):229–241, 1968.

[20] Jean-Marie Lagniez and Pierre Marquis. An improved decision-DNNF compiler. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 667–673. ijcai.org, 2017.

[21] Ken Levasseur and Al Doerr. *Applied Discrete Structures*. Lulu.com, 2012.

[22] Nils J. Nilsson. Probabilistic logic. *Artif. Intell.*, 28(1):71–87, 1986.

[23] Umut Oztok and Adnan Darwiche. A top-down compiler for sentential decision diagrams. In Qiang Yang and Michael J. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 3141–3148. AAAI Press, 2015.

[24] Tian Sang, Fahiem Bacchus, Paul Beame, Henry A. Kautz, and Toniann Pitassi. Combining component caching and clause learning for effective model counting. In *SAT 2004 - The Seventh International Conference on Theory and Applications of Satisfiability Testing, 10-13 May 2004, Vancouver, BC, Canada, Online Proceedings*, 2004.

[25] Tian Sang, Paul Beame, and Henry A. Kautz. Performing Bayesian inference by weighted model counting. In Manuela M. Veloso and Subbarao Kambhampati, editors, *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 475–482. AAAI Press / The MIT Press, 2005.

[26] Roman Sikorski. *Boolean algebras*. Springer, third edition, 1969.

[27] Alfred Tarski. *Logic, semantics, metamathematics: papers from 1923 to 1938*. Hackett Publishing, 1983.